Dense Linear-Time Correspondences for Tracking

Štěpán Obdržálek, Michal Perďoch and Jiří Matas
Center for Machine Perception
Czech Technical University Prague

Abstract

A novel method is proposed for the problem of frame-to-frame correspondence search in video sequences. The method, based on hashing of low-dimensional image descriptors, establishes dense correspondences and allows large motions. All image pixels are considered for matching, the notion of interest points is reviewed. In our formulation, points of interest are those that can be reliably matched. Their saliency depends on properties of the chosen matching function and on actual image content.

Both computational time and memory requirements of the correspondence search are asymptoticaly linear in the number of image pixels, irrespective of correspondence density and of image content. All steps of the method are simple and allow for a hardware implementation.

Functionality is demonstrated on sequences taken from a vehicle moving in an urban environment.

1. Introduction

Establishing correspondences in video streams is a widely studied problem in both computer vision and robotics. Applications include simultaneous localisation and mapping (SLAM) [14], visual odometry for autonomous vehicles (in support of GPS, wheel odometry, or accelerometers) [12, 4], motion segmentation (detection of objects of interest by motion) [8], optical flow [3], video compression, medical image registration, mosaicing, and tracking (trajectory detection) of moving objects.

Two classes of approaches to the problem can be found in the literature. The first are approaches using a gradient descent. Methods differ in optimisation procedure employed, in criterion minimised (*e.g*. sum of square differences), and in parameterisation of the frame-to-frame local transformations. Refer to [2] for more details and comparative evaluation. For example, the approach by Lucas and Kanade [11], improved later by Shi and Tomasi in [18], extracts interest points in the initial image and tracks them in consecutive frames by gradient descent.

Methods of the other class are based on detection and matching of local features or regions. They identify correspondences (motion vectors) as best matches of small spatial neighbourhoods between adjacent frames. Many approaches can be found in the literature, from the earliest works of Moravec [13] or Förstner [6] to the recent applications as in the work of Nister [14]. Good overview of the methods is given by Lepetit and Fua in [9]. In general, feature points are matched between pairs of frames using a similarity measure (*e.g*. normalised correlation) over a small window. A feature in one frame is compared to every feature within a fixed distance in the other, and mismatches are rejected by enforcing global geometry constraints.

Our application is detection of *moving objects* in scenes observed by a *moving camera*, tied to a collision avoidance and reaction planning system. This application introduces following specifics to the correspondence problem:

(i) Given the resolution of images (640x480 pixels), frame rate (10 Hz), and speed of moving objects and of the camera itself, the frame-to-frame difference in object position (the length of motion vectors) goes up to circa 40 pixels. Unless a multi-scale variant is used, gradient descent solutions fail on such large displacements by converging into a closer, false local minima of similarity.

(ii) The object detection performance increases with increasing density of correspondences. Hence, densest possible correspondences are sought, complete motion field would be ideal. This differs from the problem of visual odometry [14], where the correspondences are used only to robustly recover a 5-parametric camera motion, and tens of correspondences per frame suffice.

(iii) Since independently moving objects are observed, a guided matching, based on a global geometry of the scene, has limited use. Instead of first solving for camera motion (*e.g*. by determining epipolar geometry) from sparse correspondences, and densifying the correspondences later by guided (constrained to epipolar lines) matching, we search for dense correspondences from the beginning.

(iv) Thanks to the collision avoidance application, a solu-

*The authors were supported by Toyota Motor Corporation and by Czech Ministry of Education project 1M0567.

978-1-4244-2340-8/08/$25.00 ©2008 IEEE

tion is sought that would allow an implementation with real-time responses.

This paper presents a novel method for establishing frame-to-frame correspondences. According to the stated prerequisites, the method handles *large displacements*, produce *dense correspondences*, and its computation time is linear with respect to the number of pixels, *i.e. independent of correspondence density*.

In Section 2 the correspondence-in-video problem is analysed in detail and the method is proposed. Section 3 describes implementation particulars and Section 4 gives an experimental verification.

2. Overview of the method

We motivate our approach by establishing an analogy to the correspondence search algorithms used in object recognition and wide-baseline stereo matching. Let us first review general structure of these algorithms:

1. Decompose the image to local features – circular, elliptical or square patches around interest points, which have location and shape locally covariant with image transformations.
2. Photometrically normalise the patches.
3. Represent the patches by a misalignment-insensitive descriptor (*e.g.* SIFT [10] or DCT [16]).
4. Match patches between images, form a set of tentative correspondences (TCs). The correspondences are obtained by nearest neighbour search in the patch representation space.
5. Identify mismatches, select a subsets of TCs that are geometrically consistent with respect to *e.g.* epipolar geometry or local homography.
6. (optional) Compute dense pixel-to-pixel correspondences by guided matching. In the case of matching guided by the epipolar constraint, the dense correspondence search is restricted to search along a single line.

Let us consider how the scheme simplifies for the correspondnces-in-video problem, and what consequent simplifications to the algorithm can be made:

1. The geometric transformations between consecutive video frames can be at local scale approximated by translation (only 2 degrees of freedom). Scaling, rotation and skew are typically negligible. Consequently, circular or square patches of a fixed size and orientation can be used to form descriptors of interest points.
2. The photometric changes are small unless shadows or specular reflections are present. Hence, there is no need for photometric normalisation at local scale.
3. Compared to object recognition or wide-baseline matching problems, the set of potentially corresponding points is relatively small. The number of match candidates is limited by a predefined search-range – the maximal motion vector length. If the matches are sought in a search window of say 64×64 pixels, there are at most 4096 pixels (match candidates) to choose from. Consequently, low-dimensional descriptors, even though having low discriminative potential, are befitting. Using a high dimensional descriptor, such as the 128-dimensional SIFT [10], is unnecessary.
4. The advantage of having a low-dimensional descriptor is in the ability to organise all match candidates into a *table indexed directly by the descriptor*. Each tentative correspondence can then be retrieved in a *constant-time*.
5. If the scene contains moving objects, no global model, as *e.g.* the epipolar geometry, can be imposed on the correspondences. Mismatches are therefore identified only by requiring that, locally, multiple correspondences infer similar motion vectors.

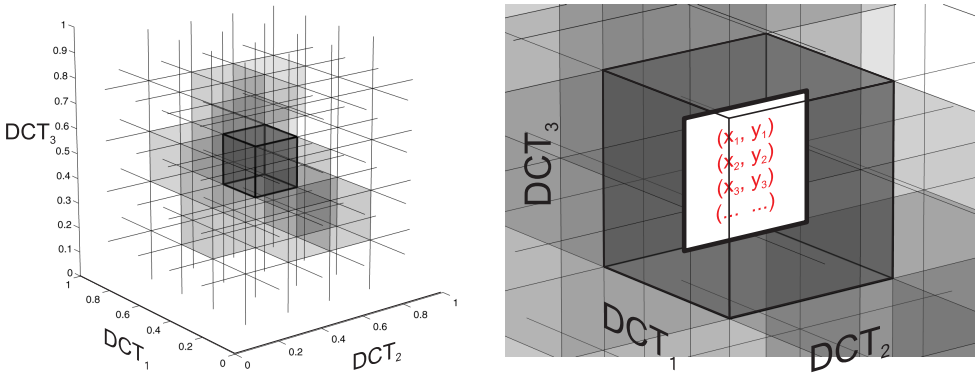
What descriptor to use, and what is the most appropriate definition of interest points in the correspondence-in-video problem? Let us start by considering the matching function first. The matching scheme in the wide baseline algorithm has quadratic $O(n^2)$ asymptotic complexity (here n is the number of interest points in the search window), and will inevitably slow down as the density of correspondences increases. We instead propose to take advantage of having a low-dimensional descriptor, and to organise match candidates into hash tables *indexed directly by values of the descriptor*. The matching function then retrieves correspondences from the table in a constant time.

Any low-dimensional descriptor that provides good discriminativity and insensitivity to noise and geometric misalignments can be used. Good examples are Gabor filters [5], low-frequency coefficients of two-dimensional discrete cosine transform [17], or other low-dimensional projections [7].

Based on the results in [15, 16], our descriptor consists of three low-frequency DCT coefficients. The DCT is computationally efficient and hardware implementations are widely available due to its widespread use in image and video compression (JPEG, MPEG, etc.). The discrete cosine transform of an image window I is defined as:

$$D_{p,q} = \alpha_p \alpha_q \sum_{m=0}^{K-1} \sum_{n=0}^{K-1} I_{m,n} \cos \frac{\pi(2m+1)p}{2K} \cos \frac{\pi(2n+1)q}{2K}, \quad (1)$$

where K is the window size in pixels, D is an output matrix of coefficients, $p : 0 \leq p \leq K-1$ and $q : 0 \leq q \leq K-1$



(a) (b)

Figure 1. (a) A hash table representing a search window, indexed by quantised DCT coefficients. (b) Coordinates of points with identical quantised descriptor listed in a cell.

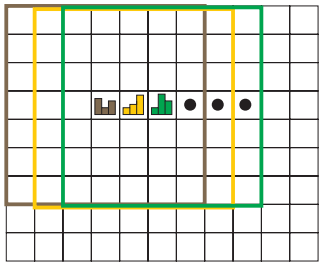


Figure 2. Three numbers describe local appearance in $K \times K$ neighbourhood of each pixel. The numbers are low-frequency DCT coefficients obtained by convolving the image with separable masks of DCT bases.

are coefficient indices, and

$$\alpha_p, \alpha_q = \begin{cases} 1/\sqrt{K} & \text{if } p \text{ resp. } q = 0 \\ \sqrt{2/K} & \text{otherwise.} \end{cases}$$

The DCT coefficients are then quantised to form a hash key (of *e.g.* 15 bits). In other words, the intensity information in a $K \times K$ window is hashed into a key, where the hash function is computed as (i) compute the low-dimensional descriptor, (ii) uniformly quantise its values, (iii) concatenate the quantised values to obtain the key.

A table is then formed for *every search window*, with cells listing coordinates of points with identical keys. See Figure 1 for an illustration. Capacity κ of each bin is limited to a low number of points, $\kappa = 3$ in our experiments. If there are more than κ points with identical key in a search window (*e.g.* there is a uniform surface in the scene), the bin capacity is saturated. The points are considered ambiguous and are not stored. Matching proceeds as follows. For a point p in frame t the key is computed. Up to κ correspondences from frame $t-1$ are retrieved from the table cell indexed by p's key.

The remaining question in the algorithm is what interest points to select for the matching. Maximising the number of correspondences, the obvious answer is all for which matches can be reliably found. *I.e.* all points that are unique within the search window, all that have an unambiguous descriptor. Which are those that remain listed in non-saturated cells of the table. In other words, there is no a-priori assumption about properties of interest points, as is *e.g.* for the Harris corner points. The only requirement is that the local $K \times K$ neighbourhood is reasonably unique. Therefore, the selection is driven solely by image-content, the *saliency of a point is directly given by uniqueness of its descriptor within the search window*.

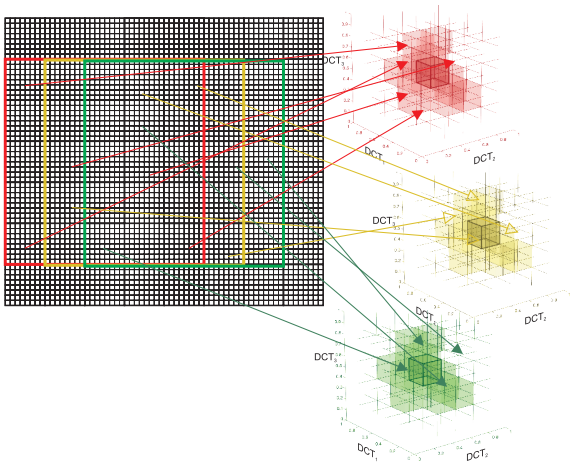


Figure 3. Video frame $t-1$: A table is crafted for each search window. The windows are placed with a granularity of 16 pixels.

3. Implementation

This section gives a step by step description of the correspondence search process.

DCT representation is computed. For every pixel in video frame $(t-1)$, three low-frequency DCT coefficients

(a) (b)

Figure 4. Coverage of an image by salient points. (a) A frame from an input video sequence. (b) Uniqueness of pixel descriptors within a search window. All pixels that are not white here are used for matching.

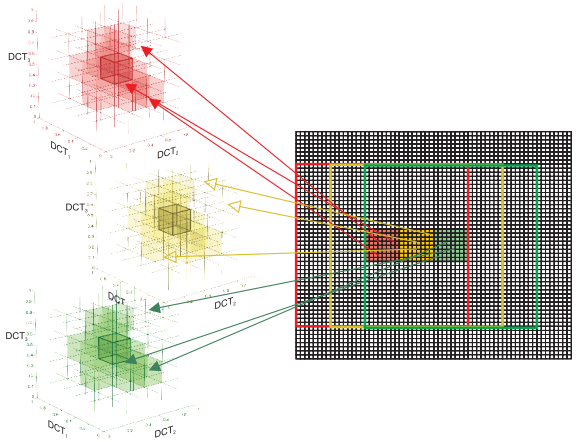


Figure 5. Video frame t: Retrieval of correspondences from precomputed tables, pixels in a 16×16 block share a single table, hence a single search window.

are computed as a dot product of image intensities in $K \times K$ ($K = 7$) windows and precomputed 2-D DCT bases. See Figure 2 for an illustration. Since the DCT bases are separable, the coefficients are obtained each by convolving the image with two one-dimensional kernels of length K. Coefficient with indices $(p, q) \in \{(1, 0), (1, 1), (0, 1)\}$ (Eqn. 1) are used. The coefficients are then discretised to 5 bits each, yielding $2^{5 \cdot 3} = 32768$ possible quantised keys. The computation has complexity comparable to that of the Harris corner detector, and is linear in the number of image pixels.

Hashing. Three-dimensional tables are created for each search window, 80×80 pixels in size. The tables list coordinates of pixels in the search window, indexed by the descriptors. For efficiency, the window position does not change with every pixel, but is rather 'jumping' over the image with a predefined granularity (16 pixels in our implementation). The situation is illustrated in Figure 3. It means, that in t^{th} frame, blocks of 16×16 pixels share the same search window (in frame $(t - 1)$). The maximal length of motion vector thus varies, from 33 to 48 pixels, according to the position of the pixel in the block.

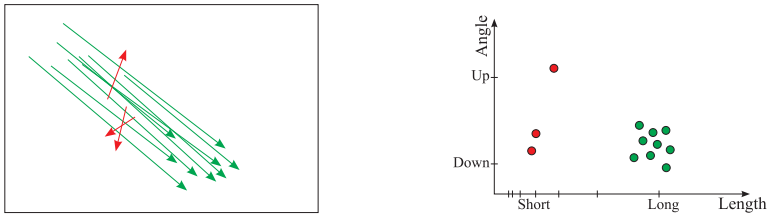
Capacity of the table cells is limited to κ ($\kappa = 3$), *i.e.* there are at most κ pixel coordinates listed for each quantisation of the descriptor. If more than κ pixels have the same descriptor, the cell is marked as saturated (an analogy to stop-lists used in text search). Salient pixels are those in non-saturated cells. Figure 4 shows an example of the saliency – uniqueness of the descriptors. Intensity of a pixel in 4(b) encodes the number of other pixels within the cell. The image is black where the descriptor is unique within the search window, white areas are where the cells are saturated.

The computational time is linear in the number of pixels. Coordinates of each pixel are entered into $5 \times 5 = 25$ tables (there are 5×5 blocks of 16×16 pixels in the 80×80 search windows, refer to Figures 3 and 5). Thanks to the limited capacity of the table cells, the tables occupy a fixed memory space.

Matching. DCTs are computed for t^{th} video frame. A pixel position identifies a search window and its associated hash table. The quantised descriptor identifies the cell, in which the pixel's correspondences are listed. See Figure 5 for an illustration. Correspondences for all pixels from *e.g.* the red block are retrieved from the red table, which was computed using pixels from the red rectangle (search window) in the previous frame. Zero to κ correspondences are retrieved for each pixel. If the indexed cell was marked as saturated, no correspondences are established due to ambiguity. Experiments show that circa 5–10% of pixels have a correspondence assigned by this procedure. The computational complexity is again linear in the number of pixels.

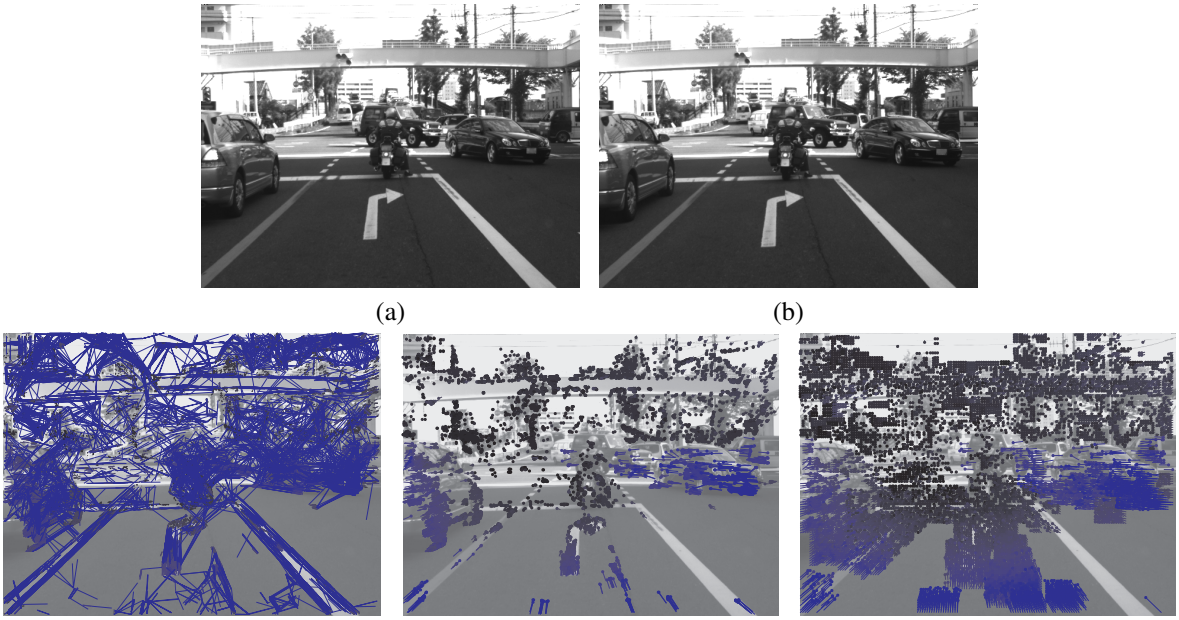
Consistency check. The set of correspondences obtained in the previous step contain a rather large number of mismatches – typically more than one half of the correspondences is incorrect. This is by design, since multiple matches per pixel are allowed.

Consistent subsets of correspondences are identified, which (a) are spatially close, and (b) represent similar mo-



(a) (b)

Figure 6. Illustration of accumulator voting in the consistency check step. (a) Tentative correspondences (motion vectors). (b) Corresponding votes in a log-polar accumulator. The red ones are rejected as inconsistent.



(a) (b)

(c) (d) (e)

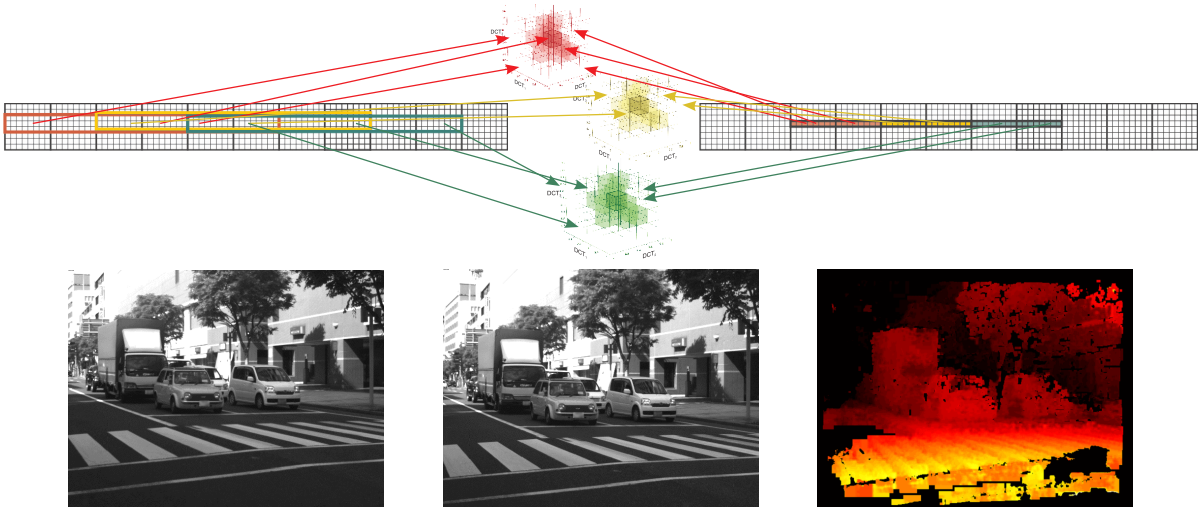
Figure 7. Progress of the correspondence search. (a), (b) Consecutive video frames from the input sequence, 640×480 pixels, (c) 22 051 tentative correspondences including many mismatches, (d) 10 205 correspondences after consistency check, (e) 194 390 dense correspondences. Only a subset, with 4×4 spacing, is displayed for presentation clarity.

tions. A Hough-transform-like approach is employed. The process has two passes. In the first pass, the motion vectors vote in a two dimensional accumulator, incrementing value in the bin representing their length and direction. In the second pass, the correspondences are checked against the accumulator, and only those that have in the corresponding bin more than Θ_c votes ($\Theta_c = 10$) are kept. See Figure 6 for an illustration. The process is linear in the number of correspondences established in the matching step.

Densification of the correspondences. As an optional step, the correspondences are propagated to neighbouring pixels where no matches were established, *i.e.* to pixels with ambiguous appearance. An approach similar to the consistency check is used. Voting in local accumulators identifies dominant motion vectors. Unmatched pixels are assigned a motion vector which got a high number of votes, given that the DCT representation of pixels being put into correspondence is similar. *I.e.* the descriptors must be similar but do not have to be unique – the ambiguity is resolved by the presence of motion vectors in close neighbourhood. After the densification, about 50% of all image pixels have a motion vector assigned. Complexity of the procedure is again linear in the number of image pixels.

Altogether, asymptotic complexity of the whole process is linear, respectively constant for a fixed image size. Figure 7 illustrates the steps of the method, showing the initial tentative correspondences (c), correspondences found to be consistent (d), and the dense correspondences (e).

With only a minor modification, the process can be applied to the problem of dense stereo matching on rectified images. There, the correspondences are sought along epipolar lines, which, by the rectification, coincide with image scanlines. Adapting the shape of the search windows by elongating them along the epipolars and shortening them in the perpendicular direction (down to a single pixel, or *e.g.* three pixels if some tolerance is required), the method



(a) (b) (c)

Figure 8. The matching procedure can be applied to dense stereo matching in rectified images. The only modification is in the shape of the search windows. (a), (b) an example of input stereo pair, (c) dense correspondences (disparity map).

provides dense stereo correspondences. The modification, as well as an example of computed dense disparity map, is shown in Figure 8.

4. Experiments

The algorithm was tested on video sequences taken in an urban environment. The results of the proposed algorithm were compared to two state-of-the-art methods. A quantitative evaluation was not performed, we only demonstrate and highlight the differences between the methods. Additionally, we present results in a visual odometry application.

4.1. Comparison to Other Methods

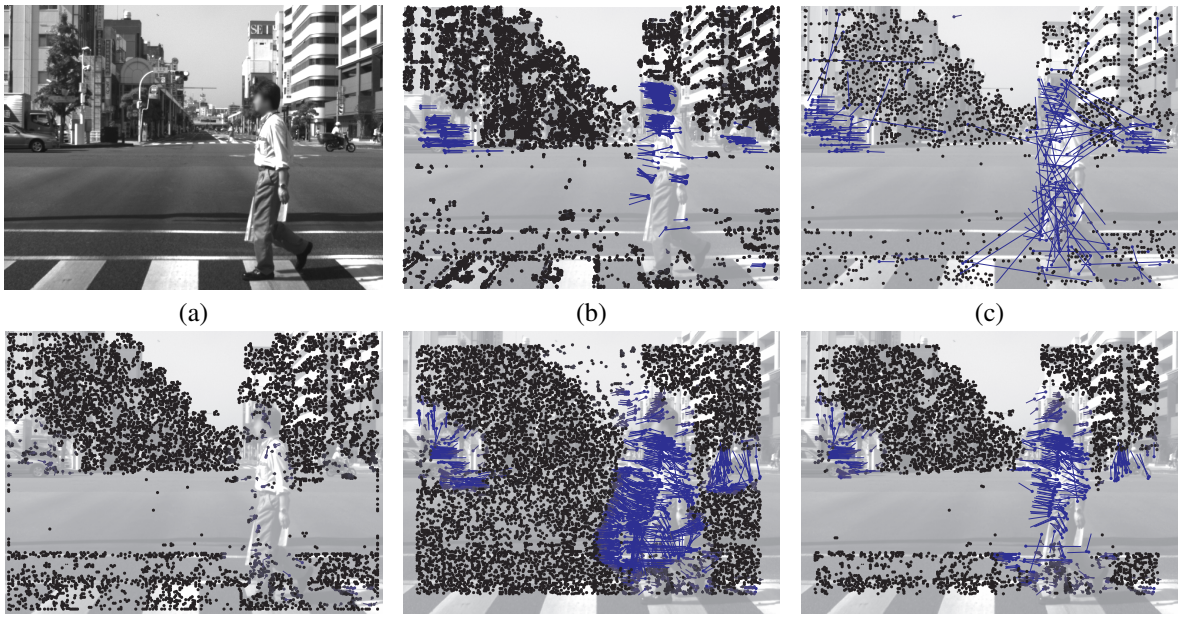
The proposed method is compared to two state-of-the-art methods for finding correspondences in video – multi-scale version of Kanade Lucas Tomasi (KLT) tracker and a method based on the correlation of rectangular patches around Harris points.

The KLT tracker, provided by S. Birchfield[1], was tested in three setups. The first setup (KLT[1]) represents the original KLT tracker without a multi-scale functionality. Second setup (KLT[2]) uses the configuration that comes with Birchfield's implementation, only the minimum distance between features was decreased to three pixels. Since there is no threshold on 'cornerness' (the lower eigenvalue of the second moment matrix of image gradients), this setup usually produced the most dense motion field of all tested methods. However, it generated a large amount of mismatches on uniform surfaces. In the last setup (KLT[3]), only such points were considered where the lower eigenvalue was above a empirically set threshold.

The correlation-based tracker closely follows the implementation described by Nister in [14]. Mutually nearest matches are established using normalised cross-correlation of 11×11 windows around feature points. The feature points were selected as local maxima of the Harris response function. No threshold was imposed on the value of the response, only extremality in 5×5 neighbourhood was required.

Figure 9 shows motion fields computed by the tested methods. The result of the proposed method is shown in Figure 9(b). The optional densification step was not applied, still the method produced the highest number of correspondences, most of them correct. Figure 9(c) shows the result of the correlation-based tracker. The motion field is sparse but again mostly correct. The number of mismatches is low, especially considering that no geometric consistency was enforced. Fig. 9(d) is for KLT[1] tracker without the multi-scale functionality. The correspondences are correct only on static parts of the scene. The moving objects cause large displacements in the image and the iterative process did not converge there. Fig. 9(e) shows results for KLT[2] with the multi-scale extension. The tracker additionally identified the large motions. Sometimes, wrong detections on a lower resolution (a higher level of a multi-resolution pyramid) caused errors from which the process on full resolution would not recover (see *e.g.* the motorcycle on the right, where most of the motion vectors point up). As there is no threshold on the 'cornerness' of the points, the correspondences cover the images very evenly. In uniform image areas with only a little information available, the motion is propagated from lower resolutions to higher. The effect is that the correspondences 'spill out' from moving objects to

[1] http://www.ces.clemson.edu/~stb/klt



(a) (b) (c)

(d) (e) (f)

Figure 9. Comparison of motion fields computed by the tested methods. The dots indicate points where correspondences were established, their colour correspond to length of the motion vectors. (a) An image from the test sequence. The pedestrian is moving to the left, the cars and the motorcycle are moving to the right. (b) The proposed method, 24890 correspondences. (c) Correlation-based tracker, 2006 correspondences. (d) KLT^1, 11059 correspondences. (e) KLT^2, 14646 correspondences. (f) KLT^3, 8229 correspondences.

their surroundings, see the walking man for an example. Finally, in Fig. 9(f) the motion field is shown for the KLT^3 variant. Requiring a minimal cornerness of the points remove the often dubious correspondences on uniform surfaces, but the propagated mismatches are still present.

To summarise, the proposed method provides frame-to-frame correspondences which are, even without the densification step, as dense as correspondences of any of the tested methods, and are as reliable as those of the correlation-based method. Further examples are shown in Figure 10.

4.2. Application: Visual Odometry

The proposed algorithm was successfully applied to the visual odometry problem. A stereo sequence of 5 000 frames was acquired by a calibrated stereo camera pair mounted on a vehicle travelling through a city. A simple egomotion estimation method provided camera translation and rotation based on correspondences between every pair of consecutive frames. The trajectory shown in Figure 11 is an aggregation of 5 000 independently computed frame-to-frame trajectory increments. Even though there is neither multi-frame feature tracking nor loop-closing involved, the accumulated error of the trajectory is rather small, indicating that high-precision egomotion increments were obtained. The method is also highly reliabile, there is not a single incorrectly computed increment (mismatch) along the sequence. Such would manifest itself as an abrupt change in the aggregated trajectory.

5. Conclusions

We have proposed a novel approach to the problem of correspondence search in video sequences. Utilising a hashing technique, the detector of interest points (regions) is avoided. The hashing both identifies distinct areas in the image and enables a constant-time matching. The hash keys are formed by quantising three low-frequency DCT coefficients, though any other low-dimensional discriminative representation of local appearance can be used. The DCTs were chosen for their properties, they are fast to compute and insensitive to misalignments and small appearance changes. The implied hash function is insensitive to noise and preserves adjacency of similar appearances.

The algorithm has asymptoticaly linear complexity with respect to the number of image pixels, the matching does not slow down with increasing density of correspondences. Computation time for the first three steps of the algorithm (descriptors, hashing and matching) is practically constant for a fixed image size, *i.e*. predictable. The time depends on the scene content only when outputing the established correspondences. Memory requirements are, except for storage of the correspondences, constant as well. All steps of the method are highly parallelisable and algorithmically simple, easily implemented in hardware.

Future work. The method has only a pixel accuracy,

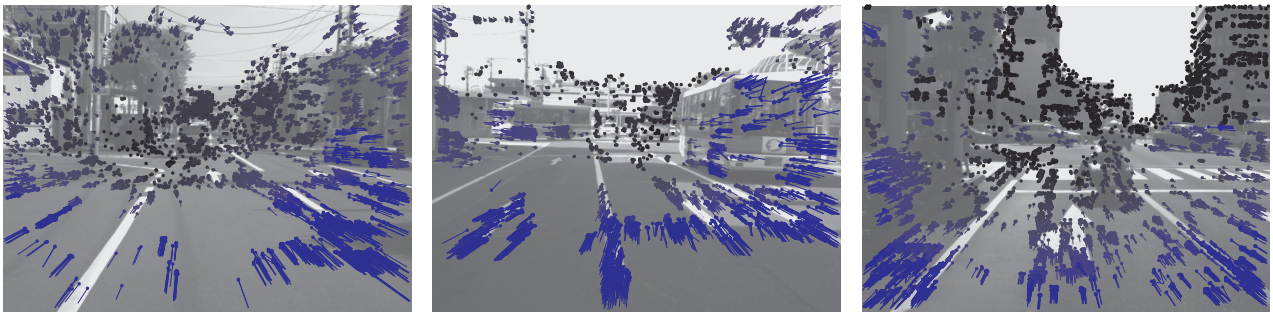


Figure 10. Examples of correspondences established by the proposed method.

making it difficult to analyse fine motions. Subpixel accuracy can be achieved *e.g.* as in the work of Anandan [1] by searching for maxima of biquadratic approximation of the pixel similarity surface.

For consecutive frames, the appearance changes are sufficiently modelled by local translations. Accommodating for deformations over longer tracks would require a model more complex than translational.

The size of the search window affects not only computation speed, but also the density of correspondences. A larger search window increases the chance that a descriptor would not be unique within the window. We consider two extensions to reduce the search window size. First is in employing a hierarchical multi-scale matching, where correspondences at a finer resolution would be sought in small neighbourhoods of correspondences at coarser scales. The other is in using a dynamic motion prediction based on tracks observed in history.

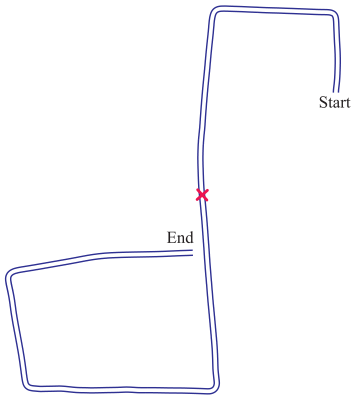


Figure 11. Visual odometry: Vehicle trajectory computed from correspondences obtained by the proposed method. The accumulated error is relatively small, the red cross marks a crossing, at which the sequence later (after circa 3000 frames) ends.

References

[1] P. Anandan. A computational framework and an algorithm for the measurement of visual motion. *IJCV*, 2(3):283–310, January 1989.

[2] S. Baker and I. Matthews. Lucas-kanade 20 years on: A unifying framework. *IJCV*, 56(3):221 – 255, March 2004.

[3] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *IJCV*, 12(1):43–77, 1994.

[4] P. I. Corke, D. Strelow, and S. Singh. Omnidirectional visual odometry for a planetary rover. In *IROS '04*, 2004.

[5] I. Fogel and D. Sagi. Gabor filters as texture discriminator. *BioCyber*, 61:102–113, 1989.

[6] W. Forstner. A feature based correspondence algorithm for image matching. In *ISPRS86*, pages III: 150–166, 1986.

[7] G. Hua, M. Brown, and S. Winder. Discriminant embedding for local image descriptors. In *ICCV '07*, Rio de Janeiro.

[8] M. Irani and P. Anandan. A unified approach to moving object detection in 2d and 3d scenes. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(6):577–589, 1998.

[9] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, 1(1):1–89, 2005.

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[11] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *IJCAI '81*, pages 674–679, April 1981.

[12] M. Maimone, Y. Cheng, and L. Matthies. Two years of visual odometry on the mars exploration rovers: Field reports. *J. Field Robot.*, 24(3):169–186, 2007.

[13] H. P. Moravec. Towards automatic visual obstacle avoidance. In *Proc IJCAI*, page 584, 1977.

[14] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *CVPR*, volume 01, pages 652–659, Los Alamitos, CA, USA, 2004. IEEE Computer Society.

[15] Š. Obdržálek. *Object Recognition Using Local Affine Frames*. Phd thesis, Center for Machine Perception, Czech Technical University, April 2007.

[16] Š. Obdržálek and J. Matas. Image retrieval using local compact DCT-based representation. In *DAGM 2003*, pages 490–497, 9 2003.

[17] K. Rao and P. Yip. *Discrete Cosine Transform: Algorithms, Advantages, Applications*. Academic Press, 1990.

[18] J. Shi and C. Tomasi. Good features to track. In *CVPR '94*, pages 593 – 600, 1994.